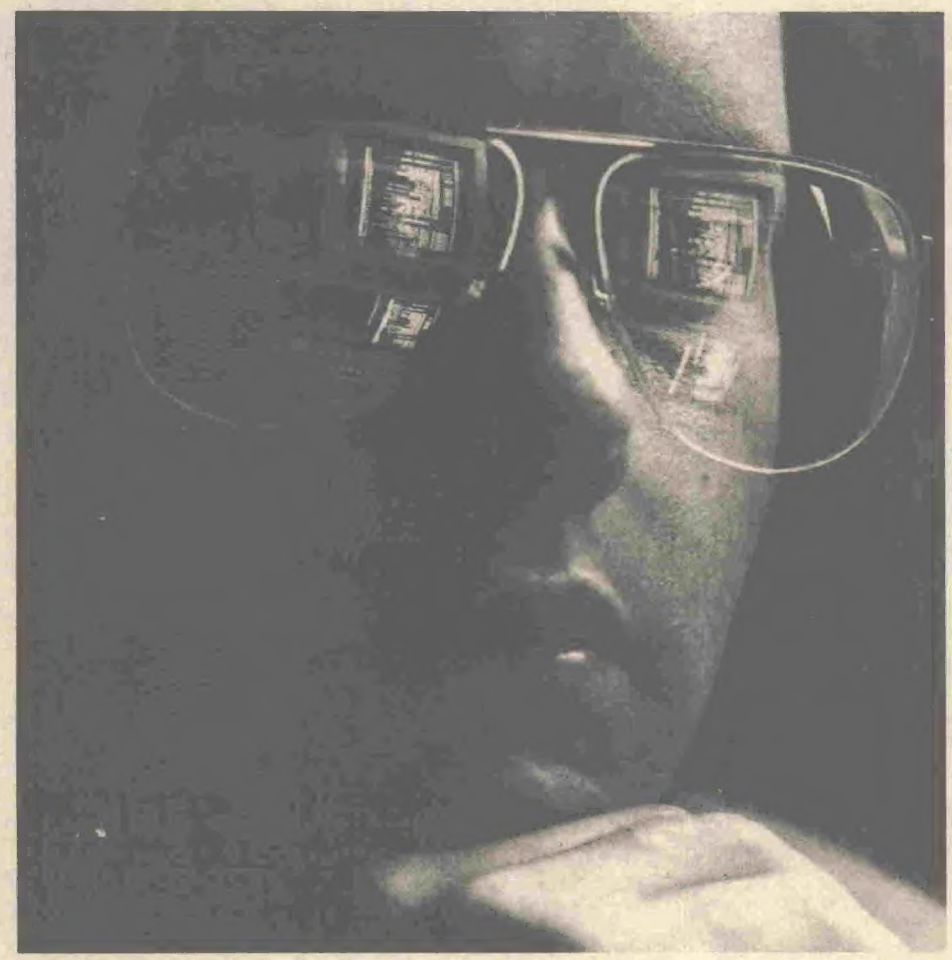
เรื่องน่าอ่าน
ดร.ดวงแก้ว สวามิภักดิ์

มุมมองของ ยูนิกซ์ (อีกมุม)



มองต่างมุมกับระบบสองระบบที่ยังแยกกันไม่ออกว่าอันไหนจะดีกว่ากัน

ผู้เขียนได้รับคำขอจากกองบก. ให้เขียนเรื่องในทำนอง "มองต่างมุม" ระหว่างระบบยูนิกซ์กับระบบแลน ซึ่งเข้าใจว่าคงจะหมายถึงแลนตระกูลโนเวลล์เน็ตแวร์โดยเฉพาะจริง ๆ แล้วคำถามในลักษณะนี้เกิดกับตัวผู้เขียนบ่อยพอสมควรในรอบ 2-3 ปีที่ผ่านมา ซึ่งก็มักจะตอบอย่างสั้น ๆ ไปว่าระบบทั้ง 2 นี้ไม่ควรนำมาเปรียบเทียบกัน มันเปรียบเสมือนอาหารจาก 2 ประเทศซึ่งเราไม่สามารถสรุปลงไปได้ว่าอาหารชนิดไหนดีกว่า ข้อสรุปนั้นย่อมต้องขึ้นอยู่กับผู้บริโภคว่ามีรสนิยมอย่างไร ? หรือมีคำจำกัดความของคำว่า "ดี" อย่างไร ? "ดี" คือคุณค่าทางโภชนาการ ? หรือ "ดี" คือความอร่อย ? (ความอร่อยของแต่ละรสนิยมก็ยังต่างกันอีก) หรือ "ดี" คือราคาถูก ? ฯลฯ ในแง่ของโอเอสก็เช่นเดียวกันไม่จำกัดอยู่ว่าจะเป็นเฉพาะระหว่างเน็ตแวร์กับยูนิกซ์ แต่ไม่ว่าจะเป็นโอเอสตัวใดนั้น เราคงสรุปลงไปเลยไม่ได้ว่า ตัวใด "ดี" กว่าตัวใด หรือแม้กระทั่งการนำมาเปรียบเทียบก็เป็นเรื่องที่ทำได้ลำบาก เพราะจะเปรียบเทียบในแง่มุมไหน ? ดังนั้นสิ่งที่กองบก.ตั้งหัวข้อให้ "มองต่างมุม" นั้น จึงเป็นเรื่องที่ถูกต้องเพราะเนื้อหาที่จะเกิดขึ้นในบทความนี้เป็นเพียง "มุมมอง" หนึ่งของ "คน ๆ หนึ่ง" และเป็นมุมมองที่เกิดจากความเห็น โดยจะไม่กล่าวถึงเนื้อหาในเชิงทฤษฎีที่สามารถหาอ่านได้จากตำรายูนิกซ์ทั่วไป

เนื้อหาของบทความนี้คงจะเน้นเฉพาะยูนิกซ์เป็นหลัก เพราะต้องขอยอมรับว่า ผู้เขียนไม่มีความรู้ความคุ้นเคยในทางปฏิบัติกับเน็ตแวร์เพียงพอที่จะวิพากษ์วิจารณ์หรือให้ทัศนะที่เป็นประโยชน์แก่ผู้อ่านได้ และผู้เขียนคิดว่าเป็นเรื่องที่ไม่ยุติธรรมอย่างยิ่งในการให้ความคิดเห็นเกี่ยวกับสิ่งที่เราไม่มีความรู้ความชำนาญเพียงพอ ดังนั้นในส่วนของเน็ตแวร์นั้นจะขอกล่าวถึงในระดับหนึ่งเท่านั้น โดยจะให้ข้อคิดเห็นเกี่ยวกับยูนิกซ์ในมุมมองของผู้เขียนเอง ซึ่งมีโอกาสได้คลุกคลีอยู่กับยูนิกซ์มาด้วยระยะเวลาพอควร

Centralized VS. Distributed

ความแตกต่างระหว่างระบบแบบรวม (centralized systems) และระบบแบบกระจาย (distributed systems) เป็นเรื่องที่น่าสนใจ และบทความที่มีผ่านตาออกมาในส่วนที่เกี่ยวกับยูนิกซ์ก็มักจะออกมาในทำนองที่ว่า "ยูนิกซ์เป็นโอเอสแบบรวม จึงไม่มีประสิทธิภาพ เพราะการประมวลผลทุกอย่างจะถูกโหลดให้เป็นภาระของเครื่องคอมพิวเตอร์ตัวกลางเสียหมด" ก่อนที่เราจะมาพิจารณาถึงรายละเอียดของข้อความนี้และนำเสนอมุมมองที่ต่างออกไป ขออนุญาตปูพื้นความเข้าใจของนิยามทางทฤษฎีของทั้งระบบรวมและระบบกระจายกันก่อน

เวลาที่เราพูดถึงระบบว่าเป็นระบบรวมหรือระบบกระจายนั้น อันที่จริงแล้วต้องมีการขยายความกันสักนิดก่อนว่าการรวมหรือการกระจายนั้นกินความถึงอะไร ? การประมวลผล (processing) และข้อมูล (data) เป็น 2 ส่วนใหญ่ที่เราต้องคำนึงถึงและแม้กระทั่งในส่วนของข้อมูลแบบกระจายเองแล้วก็ยังแบ่งระดับของการกระจายออกเป็นอีกหลายระดับไม่ว่าจะเป็นการกระจายในระดับข้อมูล ระดับไฟล์ หรือระดับฐานข้อมูล

นอกเหนือจากระบบแบบรวมและแบบกระจาย (ซึ่งก็ยังแบ่งออกเป็นอีกหลายดีกรีของการกระจายแล้ว) ก็ยังมีระบบแบบเน็ตเวอร์กที่ถือเป็น "ระบบในฝัน" ของนักคอมพิวเตอร์ทั่วไป จริง ๆ แล้วระบบเน็ตเวอร์กไม่ใช่ของใหม่ ในตำราเกี่ยวกับระบบการสื่อสารข้อมูลและตำราเกี่ยวกับระบบปฏิบัติการได้ให้คำจัดกัดความของคำว่า เน็ตเวอร์ก ไว้ว่าเป็นระบบที่มีการกระจายการประมวลผลและกระจายการเก็บข้อมูลไว้ตามโหนดต่าง ๆ โดยที่แต่ละโหนดต้องสามารถติดต่อและเรียกใช้ข้อมูลของต่างโหนดได้ด้วย ส่วนโอเอสที่จะมาควบคุมดูแลการทำงานของเน็ตเวอร์กนั้นอาจจะเป็นโอเอสตระกูลเดียวกัน (homogeneous) หรือต่างตระกูล (heterogeneous) ก็ได้

ถึงแม้ว่าแนวความคิดของเน็ตเวอร์กจะเกิดขึ้นมานานแล้ว แต่ในช่วงเวลานั้นความแพร่หลายในการใช้งานโดยเฉพาะในบ้านเรานั้นยังไม่มีมากพอ มาในปัจจุบันเมื่อเครื่องคอมพิวเตอร์มีขนาดและราคาถูกลง การนำเอาระบบหลาย ๆ ระบบมาต่อเชื่อมเข้าด้วยกัน และขนานนามว่าเป็นระบบเน็ตเวอร์กจึงมีให้เห็นกันทั่วไป และระบบที่เป็นที่นิยมและมีจำนวนการใช้กันมากที่สุดระบบหนึ่งก็เห็นจะได้แก่การใช้เน็ตแวร์ของโนเวลล์เป็นโอเอสควบคุมการทำงานของเน็ตเวอร์ก ซึ่งมักจะเรียกกันโดยปริยายไปเลยว่าเป็นระบบ "แลน" หรือ local area network มาถึงตรงนี้ก็ใคร่ขอย้ำอีกสักครั้งว่าการที่มีคนจำกัดความ คำว่าแลนว่าเป็นการนำเอาเครื่องพีซีมาต่อกันแล้วอุปโหลกเครื่องหนึ่งให้เป็นเซอร์ฟเวอร์แล้วก็ติดตั้งเน็ตแวร์ลงไปนั้น เปรียบเสมือนกับการเรียกเครื่องถ่ายเอกสารว่าเครื่องซีร็อกซ์ หรือเรียกผงซักฟอกว่าแฟ็ป นั่นเอง เพราะเน็ตแวร์นับเป็นโอเอสที่มีการทำงานแบบกระจายการประมวลผลระหว่างเครื่องแบบหนึ่งเท่านั้น จริง ๆ แล้วการวางระบบเน็ตเวอร์กนั้นยังมีทางเลือกแบบอื่นอีกมาก

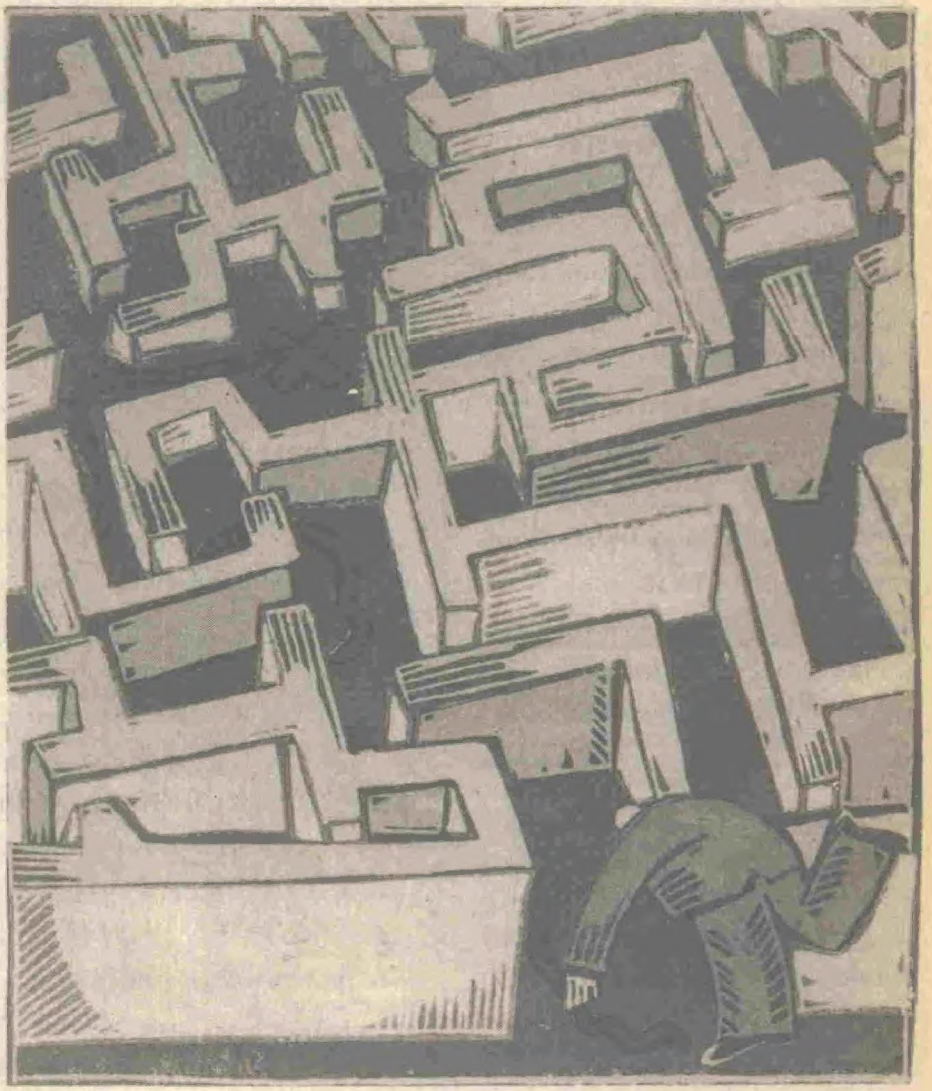
และก็ไม่ได้จำกัดว่าถ้าเป็นแลนแล้วจะต้องเป็นเครื่องพีซีเท่านั้นด้วยในฉบับหน้าเราจะได้กล่าวถึงการใช้ยูนิกซ์ในแลนกัน แต่ตอนนี้ของกล่าวถึงเน็ตแวร์กันสักเล็กน้อย

เน็ตแวร์เป็น NOS จริงหรือ ?

คงเป็นที่ยอมรับว่าปริมาณการใช้งานของสิ่งที่เราเรียกว่าเน็ตแวร์แลนนั้น มีสูงกว่าการใช้ระบบการกระจายประเภทอื่น ทั้งนี้ก็เนื่องมาจากฐานการใช้งานของดอสบนเครื่องพีซีซึ่งมีอยู่มากมาย เมื่อหน่วยงานที่เคยใช้เครื่องพีซีกันมาสักระยะหนึ่งได้เห็นประโยชน์ของเครื่องคอมพิวเตอร์ ก็เริ่มจะมองเห็นถึงความจำเป็นในการเชื่อมพีซีที่มีอยู่ในหน่วยงานเข้าด้วยกันเพื่อเป็นการแชร์ทรัพยากรของระบบ ผู้เขียนเข้าใจว่านี่คือที่มาถึงการเปรียบมวยระหว่างเน็ตแวร์กับยูนิกซ์ เพราะวิธีการที่จะเชื่อมต่อเครื่องพีซีให้ทำงานร่วมกันได้ที่สะดวกที่สุดคือการจัดหาเครื่องพีซีใหญ่ ๆ สักเครื่องให้มาทำหน้าที่เป็น host เสียทีนี้ก็มาถึงประเด็นว่าจะเอาซอฟต์แวร์จัดการอะไรใส่เข้าไปใน host ดี ซึ่งก็คงได้รับคำตอบว่าถ้าไม่เป็นเน็ตแวร์ก็ใช้ยูนิกซ์ คำถามที่ตามมาติด ๆ ก็ย่อมจะได้แก่คำถามที่ว่าระหว่างเน็ตแวร์กับยูนิกซ์อะไรจะดีกว่า อันเป็นที่มาของบทความหลาย ๆ บทความที่ออกมาวิพากษ์วิจารณ์ ซึ่งมักจะเป็นถล่มยูนิกซ์กันเสียเป็นส่วนใหญ่ มาถึงตรงนี้ขอทำความเข้าใจกันอีกครั้งว่า การนำพีซีหรือที่เราอาจจะเรียกว่าสถานีงาน (station) หรือโหนดหรืออะไรก็แล้วแต่มาเชื่อมต่อกันนั้นมีทางเลือกในการติดตั้งมากมาย ไม่ว่าจะเป็นรูปแบบของเครือข่ายโปรโตคอล ฯลฯ และในส่วนของโอเอส

นั้นก็ไม่ได้มีเฉพาะเน็ตแวร์หรือยูนิกซ์เท่านั้น โอเอสที่ใช้ในการบริหารเน็ตเวอร์กที่มีพีซีทำงานภายใต้ดอสเป็นสถานีงานนั้น นอกจากเน็ตแวร์แล้วยังมีอีกเป็นสิบ ๆ ชนิด และก็เช่นเดียวกันคือ ยูนิกซ์ไม่ได้ทำงานเฉพาะกับ host ที่เป็น Intel based หรือที่เราเรียกกันว่าพีซีเท่านั้น ดังนั้นการจะนำเน็ตแวร์มาเปรียบเทียบกับยูนิกซ์นั้นจึงเป็นเรื่องที่ไม่น่าจะหาข้อสรุปได้

ว่าจะพูดถึงเน็ตแวร์ก็มัวแต่เปะปะเรื่องอื่นอยู่ เอาล่ะ ! ขอเข้าเรื่องเน็ตแวร์ที่ว่าจะพูดถึงกันเสียที อย่างที่ได้ออกตัวถึงความไม่ประสีประสากับเน็ตแวร์เอาไว้แล้ว ก็ขอสรุปประเด็นที่คิดว่าน่าสนใจเอาไว้ให้เข้ากับหัวข้อคือ การทำงานของเน็ตแวร์นั้นถ้าจะว่ากันตามทฤษฎีแล้วจะไม่สามารถเรียกตัวเองว่าเป็นระบบเน็ตเวอร์กที่แท้จริงได้ เพราะการจัดเก็บข้อมูลของเน็ตแวร์นั้นเป็นแบบรวม คือข้อมูลที่แชร์ใช้อยู่ในระบบนั้นจะต้องมาจัดเก็บที่ host สิ่งที่ตามมาก็คือสถานีงานแต่ละตัวไม่สามารถจะเรียกใช้ข้อมูลที่อยู่ในสถานีงานตัวอื่นได้ นอกจากนี้ระบบไฟล์ของเน็ตแวร์นั้นยังไม่โปร่งใส (transparent) ต่อผู้ใช้ คือผู้ใช้ต้องระบุเป็นเรื่องเป็นราวด้วยซื้อไดรฟ์ว่าต้องการไฟล์ส่วนกลางหรือกำลังใช้ไฟล์ในเครื่องของตนเท่าที่ได้เคยติดตามดูก็เห็นจะมีซอฟต์แวร์ที่ชื่อ Banyan ของ Vines และ LANtastics ของ Artisoft ที่บริหารข้อมูลแบบกระจายจริง ๆ โดยสรุปก็คือเน็ตแวร์จัดเป็นโอเอสของเน็ตเวอร์กที่ทำการประมวลผลแบบกระจาย แต่จัดเก็บข้อมูลแบบรวม นั่นก็คือ ถ้าเรามีความเชื่อมั่นในความเร็วของการเรียกใช้ข้อมูลของเครื่องที่ศูนย์กลางและไม่ติดขัดกับปัญหาในการควบคุมภาวะของการเข้าหาข้อมูลพร้อม ๆ กัน (concurrency control) แล้วล่ะก็ เน็ตแวร์ก็นับเป็นทางเลือกทางหนึ่งที่เป็นไปได้



หันมามองยูนิกซ์

กลุ่มผู้ที่ชอบนำยูนิกซ์มาเปรียบเทียบกับเน็ตแวร์มักจะสรุปว่ายูนิกซ์มีจุดอ่อนที่เป็นประเด็นหลัก ๆ อยู่ 2 เรื่อง ได้แก่ปัญหาในแง่ของความเร็ว โดยเฉพาะความเร็วในการโต้ตอบการใช้งาน (response time) ส่วนอีกเรื่องหนึ่งก็คือการใช้งานที่ (คิดว่า) ยุ่งยาก ในเรื่องนี้เราอยากจะพิจารณาถึงประเด็นทั้ง 2 นี้รวม ๆ กันไปเลย

ว่ากันว่ายูนิกซ์นั้นใช้ยาก และไม่ค่อย friendly กับผู้ใช้ ตรงนี้ขอให้มองเจาะกันไปเลยว่าเรากำลังพูดถึงผู้ใช้ประเภทใด และลักษณะการใช้นั้นใช้อย่างไร ? ซึ่งถ้าจะแบ่งกันไปแล้วประเภทของผู้ใช้จะมีอยู่ 2 กลุ่มคือ

- ผู้ใช้โปรแกรมประยุกต์
- ผู้พัฒนาโปรแกรม

ยูนิกซ์กับผู้ใช้โปรแกรมประยุกต์

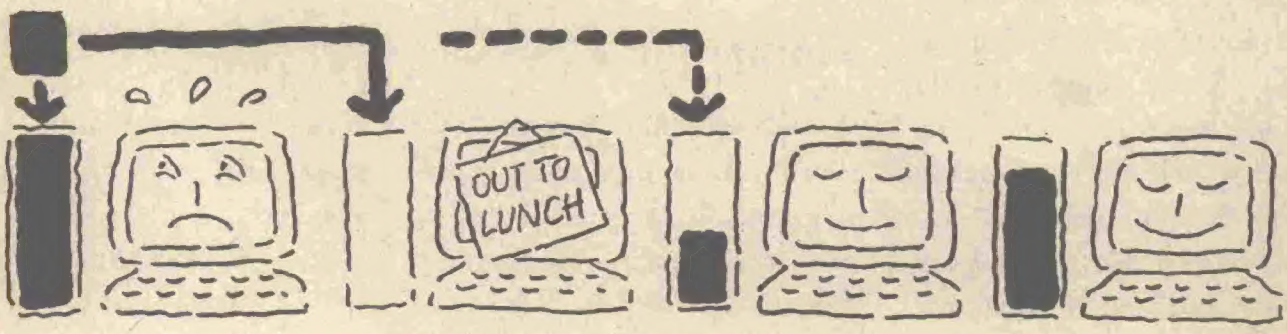
ผู้ใช้ประเภทแรกนั้นก็คือผู้ใช้จริง ๆ อันได้แก่ ผู้ที่ใช้โปรแกรมที่ถูกพัฒนาขึ้นมาจริง ๆ แล้วผู้ใช้ประเภทนี้ไม่ต้องสนใจด้วยซ้ำว่ากำลังใช้โอเอสตัวใดอยู่ เพราะรูปแบบของงานที่ใช้อยู่ส่วนใหญ่ก็มักจะเป็นเมนูให้เลือก และคีย์ข้อมูลลงในช่องที่กำหนดเสียมากกว่า จุดที่ผู้ใช้กลุ่มนี้สนใจคือความถูกต้องสมบูรณ์ และระดับความเร็วของการโต้ตอบ รวมถึงความสวยงามของจอภาพ และรูปแบบรายงาน มาถึงตรงนี้ ผู้ที่เคยใช้ยูนิกซ์บางท่านก็อาจจะนึกอยู่ทันทีว่า นี่แหละคือจุดอ่อนของยูนิกซ์ เพราะโปรแกรมที่พัฒนาขึ้นภายใต้ยูนิกซ์มักจะขาดคุณสมบัติของความเร็ว และถูกเล่นของหน้าจอ ซึ่งขอท้าวความเดิมที่ได้พูดถึงเมื่อตอนต้นของบทความว่า ความช้าของยูนิกซ์นั้นมีสาเหตุมาจากการที่งานทุกอย่างถูกโหลดไปให้ซีพียูส่วนกลางเป็นผู้กระทำเสียหมดนั้นมีข้อเท็จจริงประการใด จริง ๆ แล้วโปรแกรมจะทำงานได้มีประสิทธิภาพเพียงใดนั้นขึ้นอยู่กับฝีมือและประสบการณ์ของผู้พัฒนา ก็เลยต้องขอออกจากส่วนของผู้ใช้โปรแกรมประยุกต์เข้าสู่การใช้งานของผู้พัฒนาโปรแกรมกันเลย

ยูนิกซ์กับผู้พัฒนาโปรแกรม

ในส่วนของยูนิกซ์กับผู้พัฒนาโปรแกรมนั้นจริง ๆ แล้วน่าที่จะมีความสัมพันธ์ที่ดีต่อกัน เพราะปรัชญาประการหนึ่งของยูนิกซ์คือเป็นโอเอสที่สร้างขึ้นสำหรับโปรแกรมเมอร์ และสร้างด้วยฝีมือของโปรแกรมเมอร์ระดับปรมาจารย์ ผู้ที่ได้ใช้ยูนิกซ์โดยเฉพาะเมื่อใช้ในงานพัฒนาโปรแกรมมาถึงระดับหนึ่งมักจะยอมรับในความสะดวกของบรรดาเครื่องไม้เครื่องมือที่ยูนิกซ์มีไว้ให้ใช้อย่างครบครัน โดยไม่ต้องซื้อต้องหา ในส่วนที่ว่า "โดยไม่ต้องซื้อต้องหา" นั้นนับว่ามีความสำคัญทีเดียว เพราะนั่นก็หมายความว่า บรรยากาศในการทำงาน ไม่ว่าจะเป็นการทำงานบนคอมพิวเตอร์เครื่องใด ยี่ห้อใด ถ้าระบบเหล่านี้ใช้ยูนิกซ์แล้ว ในแง่ของผู้พัฒนาโปรแกรมก็จะเจอเครื่องมือเหล่านี้แทบจะเหมือนกันหมด เราไม่ต้องคอยพกดิสก์ที่บรรจุซอฟต์แวร์ที่เราใช้ประจำติดตัว และต้องเสียเวลาติดตั้งใหม่เมื่อต้องมาใช้เครื่องหรือระบบที่ไม่มีซอฟต์แวร์นั้น ๆ ติดตั้งเอาไว้ เพราะในแง่ของคอสนั้น ความหรูหราของเครื่องมือต่าง ๆ มาจากโปรแกรมเสริมจากบริษัทผู้ผลิตต่าง ๆ ความสามารถเหล่านี้ไม่ได้อยู่ในส่วนของโอเอส และแม้กระทั่งคอสเองก็มีหลากหลายเวอร์ชัน และแต่ละเวอร์ชันก็มีความแตกต่างในแง่ของเครื่องมือซอฟต์แวร์มากเหลือเกิน สรุปตรงนี้ก็คือ เป็นที่ยืนยันได้ว่ายูนิกซ์มีสิ่งดี ๆ ให้เหล่าผู้พัฒนาโปรแกรมใช้อย่างสะดวกสบาย ข้อติดขัดในส่วนนี้ก็คือ เหล่าโปรแกรมเมอร์มักจะไม่ยอมสละเวลาในการเรียนรู้การนำเครื่องมือเหล่านี้มาใช้ เพราะถ้าจะมองว่ามีแค่เอดิเตอร์ กับคอมพายเลอร์ก็เขียนโปรแกรมได้แล้วนั้น มันก็เหมือนกับการใช้เข็มกับด้ายเย็บชุดดินเนอร์หรู ๆ นั่นแหละ อย่างไรก็ดีเนื้อหาในส่วนของยูนิกซ์กับการพัฒนาโปรแกรมนั้นยังมีรายละเอียดอีกมาก ซึ่งถ้ามีโอกาสคงจะได้นำข้อมูลในส่วนนี้มาเสนอเป็นเรื่องต่างหากอีกที

มาจัดระบบให้มีประสิทธิภาพ

เอาล่ะ สมมติว่าเรายอมรับแล้วว่ายูนิกซ์มีเครื่องมือดี ๆ ที่ช่วยในการพัฒนาโปรแกรมให้เราได้ แต่อย่างไรก็ตาม การใช้งานของยูนิกซ์ (ระบบเดียว) ก็ยังเป็นการใช้งานแบบรวมทั้งในแง่การประมวลผล และในส่วนของข้อมูล และถ้ามองตามหลักการแล้วระบบแบบนี้ถ้ามีจำนวนผู้ใช้เข้ามาเยอะ ๆ แล้ว ระบบก็ย่อมจะหนืดอย่างแน่นอน ที่กล่าวมานี้นับเป็นความจริงแต่มันจะเป็นจริงเฉพาะเมื่อเราไม่ได้พยายามทำอะไรเลย การทำให้คอมพิวเตอร์ทำงาน



ให้ได้อย่างมีประสิทธิภาพนั้นเป็นทั้งศาสตร์และศิลป์ของเหล่าผู้พัฒนาโปรแกรมที่ต้องให้ความสนใจและติดตามศึกษารวมทั้งนำมาปฏิบัติ ในที่นี้เรากำลังพุ่งประเด็นไปที่ความช้าของยูนิกซ์ที่ทำงานแบบ time sharing โดยโปรแกรมทั้งปวงในระบบจะมาโหลดให้ host ประมวลผล และยูนิกซ์จะแบ่งเวลาการใช้ซีพียูให้แก่โปรเซสเหล่านี้ผู้ที่ได้ศึกษาทฤษฎีของระบบปฏิบัติการคงพอจำได้ว่า ลักษณะงานที่เข้ามาอยู่ในระบบนั้นสามารถแบ่งได้เป็น 2 ประเภทคือ งานแบบ CPU bound และงานแบบ I/O bound งานแบบ CPU bound นั้นตามชื่อก็บอกอยู่แล้วว่าเป็นงานประเภทที่จะใช้เวลาของซีพียูมาก กล่าวคือ ในส่วนของโปรแกรมนั้นจะประกอบด้วยส่วนของการคำนวณและการเปรียบเทียบเป็นหลักใหญ่ ไม่ค่อยมีการรับ-ส่งข้อมูล ส่วนงานแบบ I/O bound นั้นจะอ่าน-เขียนข้อมูลกันเป็นหลัก และถ้าเราลองวิเคราะห์กันต่อไปก็สรุปได้คร่าว ๆ ว่างานทางด้านคาด้าโปรเซสซิงจัดเป็นงาน I/O bound ส่วนงานประเภท CPU bound ได้แก่ การทำงานทางด้านกราฟิก การวิเคราะห์ทางสถิติ และการคอมพายล์โปรแกรมเป็นต้น อันนี้ผู้เขียนเห็นด้วย 100% ว่าถ้าเราเอางานทุกประเภทที่กล่าวมานี้มาวิ่งบนยูนิกซ์ระบบเดียวกันและแถมยังมีผู้ใช้จำนวนมาก ระบบย่อมรับไม่ไหวแน่ แต่อย่างที่ได้เกริ่นไว้แล้วว่าการปรับระบบให้มีประสิทธิภาพนั้น เป็นหน้าที่ของนักคอมพิวเตอร์ ดังนั้นการรวมเอางาน 600 ประเภทและแถมไม่มีอะไรเกี่ยวข้องกันอีกมาอัดอยู่ในระบบเดียวนั้นเป็นเรื่องที่ควรหลีกเลี่ยงจริง ๆ เนื่องจากยูนิกซ์สามารถทำงานได้กับเครื่องมากมายหลายรุ่น

ดังนั้นการจัดให้การพัฒนาโปรแกรมที่เป็นงานแบบ CPU bound ทำออกไปต่างหากบนเครื่องพีซีสักเครื่องที่วิ่งยูนิกซ์และเอางานที่มีการประมวลกราฟิกมาก ๆ ให้ทำอยู่บนเวอร์กสเตชันที่มีความละเอียดของจอภาพสูง ๆ และวิ่งยูนิกซ์ ส่วนงานคาด้าโปรเซสซิงซึ่งคงต้องมีผู้ใช้พร้อม ๆ กันหลาย ๆ คนก็วิ่งบนเครื่องอีก 1 ระบบไปเลย ก็จะทำให้การใช้งานของซีพียูที่ host เป็นไปอย่างมีระบบและระบบที่กำลังใช้ยูนิกซ์อยู่ทั้งหมดนี้ยังสามารถต่อเป็นเน็ตเวอร์กเพื่อแชร์ทรัพยากรกันได้อีก เราจะได้กล่าวถึงเนื้อหาในส่วนของยูนิกซ์กับเน็ตเวอร์กในตอนหน้า โดยจุดที่เราอยากเน้นตรงนี้ก็คือ ผลของการจัดระบบในลักษณะนี้จะทำให้การทำงานของยูนิกซ์ที่มีผู้ใช้หลาย ๆ คนพร้อมกันนั้นเป็นงานประเภท I/O bound ทั้งหมด ซึ่งในส่วนต่อไปนี้เราจะให้ความสนใจกับกรณีนี้โดยเฉพาะกับประเด็นที่ได้พูดถึงคือในแง่ความเร็วกับความสวยงามของจอภาพ

ยูนิกซ์กับการรับ-ส่งข้อมูล

พูดถึงความเร็วของงานประเภท I/O bound แล้วก็ต้องนึกถึงจุดที่เกี่ยวข้องกับ I/O ซึ่งสามารถแบ่งออกได้เป็น 2 ส่วน คือส่วนของข้อมูลที่จัดเก็บในระบบ และ ส่วนของข้อมูลที่นำมาแสดงผล

สื่อที่ใช้เก็บข้อมูลในระบบที่เป็นที่นิยมมากที่สุดตอนนี้ก็เห็นจะได้แก่ ดิสก์ และหน่วยความจำตรงนี้คงไม่ต้องถกเถียงกันว่าการเรียกใช้ (อ่าน) หรือแก้ไข (เขียน) ข้อมูลที่อยู่ในหน่วยความจำจะเร็วกว่าในดิสก์มาก ดังนั้นก็คงจะอยู่ที่การออกแบบและการจัดระบบของโปรแกรมให้ข้อมูลที่ถูกเรียกใช้บ่อย ๆ มาอยู่ในหน่วยความจำเสีย แต่การใช้หน่วยความจำนั้นนอกจากการที่มีเนื้อที่จำกัด และการที่ไม่สามารถเก็บข้อมูลไว้ได้อย่างถาวรแล้ว สำหรับโปรแกรมเมอร์มือใหม่จะมีปัญหาในการให้หลาย ๆ โปรเซสแชร์ข้อมูลที่อยู่ในหน่วยความจำในยูนิกซ์เองมีเทคนิคของการสื่อสารระหว่างโปรเซส (inter-process communications) ให้เลือกใช้หลายแบบ และแม้กระทั่งการเขียนโปรแกรมให้เรียกใช้ไฟล์ตามปกติ แต่กำหนดให้ไฟล์นั้นมาเก็บอยู่ในหน่วยความจำ (in-memory filesystem) อันเป็นแนวความคิดคล้าย ๆ กับ RAM-disk ก็เป็นอีกเทคนิคหนึ่งที่น่าสนใจและสามารถดึงเอามาใช้ในการพัฒนาโปรแกรมได้อย่างได้ผล ซึ่งรายละเอียดเหล่านี้ก็หวังว่าจะได้มีโอกาสนำเสนอกันต่อไป อย่างไรก็ดี เราคงไม่สามารถจัดการให้ข้อมูลมาอยู่ในหน่วยความจำได้ทั้งหมด ดังนั้นการเลือกดิสก์ที่มีความเร็วในระดับที่ต้องการ การนำตัวควบคุมดิสก์ (disk controller) มากกว่า 1 ตัวมาให้ทำงานช่วยกัน ก็เป็นอีกวิธีที่จะเร่งความเร็วได้จริง ๆ แล้วในเรื่องของการเรียกใช้ข้อมูลนี้ หากจะเปรียบเทียบระหว่างยูนิกซ์กับเน็ตแวร์แล้วก็ล้วนเก็บข้อมูลแบบรวม

เหมือนกัน แต่ยูนิกซ์จะได้เปรียบที่มีความหลากหลายของฮาร์ดแวร์ให้เลือกใช้มากกว่า

ทีนี้มาถึงการแสดงผล ซึ่งเท่าที่ได้อ่านบทความอื่น ๆ มานั้นมักจะเปรียบเทียบยูนิกซ์กับเน็ตแวร์ที่ความเร็วของสายสื่อสารระหว่างตัว host กับสถานีงานว่ายูนิกซ์ต้องต่อผ่าน RS-232 ที่มีความเร็วสูงสุดเพียงแค่ 19.2 กิโลบิตต่อวินาที ซึ่งไม่ทันอกทันใจอะไรทำนองนี้จริง ๆ แล้วการต่อในระบบยูนิกซ์ไม่ได้บังคับว่าจะต้องต่อผ่าน RS-232 เท่านั้น แต่ถ้าเราจะจำเพาะเจาะจงลงให้เป็นอย่างนั้น โดยถ้ามองตัวเลขของความเร็วสายว่าประมาณ 2000 ไบต์ต่อวินาทีนั้นจะเห็นว่าเราสามารถแสดงผลเต็มหน้าจอได้ในเวลา 1 วินาที หรือถ้าจะแสดง 2 หน้าจอก็ต้องใช้เวลา 2 วินาทีนั้น ก็ต้องถามต่อว่าขนาดของข้อมูลที่จะส่งมาแสดงผลในแต่ละครั้งสำหรับงานประเภทดาต้าโปรเซสซิงซึ่งส่วนใหญ่ทำงานผ่านทางเมนูนั้นจะใหญ่สักเท่าใด คำตอบก็คือใหญ่เท่าขนาดของฟิลด์หรือเรคอร์ดซึ่งใน 2000 ไบต์ที่มีสิทธิปรากฏบนหน้าจอนั้น จริง ๆ แล้วจะเป็นข้อมูลเพียงส่วนเดียวเท่านั้น ซึ่งถ้าคำนวณออกมาแล้วการแสดงผลในแต่ละคราวน่าจะอยู่ภายใน 1 วินาทีอันเป็นความเร็วที่น่าจะยอมรับได้ ที่ว่ากันว่ายูนิกซ์ช้าเมื่อมีผู้ใช้เยอะ ๆ นั้นน่าจะเป็นความรู้สึกหนึดเวลาที่คีย์แก้ไขข้อมูล เพราะ (ถ้าไม่ได้มีการเซ็ตเป็นอย่างอื่นแล้ว) ข้อมูลแต่ละตัวที่เราคีย์เข้าไปนั้นจะส่งผ่านสายไปให้ host และส่งมา echo ที่หน้าจอเราอีกทีหนึ่ง ถ้าเป็นสมัยก่อนก็คงต้องแนะนำว่าต้องวางระบบให้จำนวนผู้ใช้พอดีกับกำลังของระบบ แต่มาในสมัยนี้เราสามารถนำพีซีมาใช้เป็นเทอร์มินัล

ของระบบได้ อันนี้ก็อยู่ที่ความสามารถของซอฟต์แวร์ที่วิ่งบนพีซีแล้วว่าจะทำให้พีซีแปลงสภาพเป็นเทอร์มินัลที่มีความเก่งในระดับใด หลาย ๆ ท่านคงคุ้นเคยกับซอฟต์แวร์ประเภท emulator ที่จะแปลงสภาพพีซีให้กลายเป็นเทอร์มินัลของยูนิกซ์ และก็คงอึดอัดกับความช้าในการทำงาน โดยเฉพาะเมื่อเปรียบเทียบกับการใช้พีซีเครื่องเดียวกันนี้เป็นสถานีงานของเน็ตแวร์ จริง ๆ แล้วความช้าที่มักเข้าใจว่าเกิดขึ้นเพราะ host ประมวลผลไม่ทันนั้นไม่ใช่สาเหตุหลักตามที่เราได้ชี้แจงถึงความแตกต่างระหว่าง CPU bound และ I/O bound ไปแล้ว จริง ๆ แล้วความช้าเกิดจากการที่ emulator ที่ทำงานอยู่บนพีซีนั้นไม่ได้ใช้ความสามารถใด ๆ ของพีซีนอกเหนือจากการรับส่งข้อมูล จากประสบการณ์ของตัวเองพบว่าเราสามารถสร้างซอฟต์แวร์บนพีซีให้กลายเป็นเทอร์มินัลที่มีความฉลาดขึ้นได้ นั่นก็คือในการทำงานนั้นซอฟต์แวร์ที่ว่าจะเข้ามาจัดการแก้ไขข้อมูลรวมทั้งจัดรูปแบบของหน้าจอให้มีถูกเล่นได้ต่าง ๆ ได้ตามความสามารถของพีซี และเมื่อแก้ไขจนพอใจแล้วจึงค่อยส่งให้ host ประมวลผล ซึ่งถ้าจะว่าไปแล้วการพัฒนาโปรแกรมในลักษณะนี้จัดเป็นการประมวลผลแบบกระจายในระดับหนึ่งนั่นเอง

เน็ตเวอร์กกับยูนิกซ์

การเปรียบเทียบยูนิกซ์กับแลนนั้นยังไงก็เป็นเรื่องแปลกในสายตาผู้เขียน เพราะมันเหมือนกับพูดว่ายูนิกซ์ต่อให้เป็นเน็ตเวอร์กไม่ได้ ทั้ง ๆ ที่ความสามารถในการทำงานแบบเน็ตเวอร์กของยูนิกซ์นั้นมีเหลือล้นพ้นประมาณ ในฉบับต่อไปเราจะยังมีมุมมองที่เกี่ยวกับยูนิกซ์กับการใช้งานแบบเน็ตเวอร์กที่ใคร่จะนำเสนอ และรอพบกันใหม่ค่ะ

หมายเหตุ : ในส่วนที่ได้กล่าวถึงเนตแวร์ว่าไม่สามารถทำงานแบบ peer to peer ได้นั้นอันที่จริงมีเน็ตแวร์อยู่เวอร์ชันหนึ่งคือ Netware Lite ที่ได้ออกแบบให้เป็นแบบ peer to peer แต่เหมาะสำหรับเน็ตเวอร์กขนาดเล็ก อย่างไรก็ดีเจ้า Netware Lite นี้จะต่างจากเน็ตแวร์ปกติคือ มันจะต้องทำงานภายใต้ดอสอีกที และใช้โครงสร้างไฟล์แบบดอสซึ่งมีประสิทธิภาพด้อยกว่าของเน็ตแวร์เองมาก โดยสรุปสำหรับผู้ใช้เน็ตแวร์แล้วส่วนใหญ่จะยังคงใช้หลักการของ centralized data ที่ใช้อยู่ในเน็ตแวร์ทั่วไปอยู่ดี